



# Biologically-Inspired Control for Multi-Agent Self-Adaptive Tasks

## Citation

Yu, Chih-Han, and Radhika Nagpal. 2010. Biologically-inspired control for multi-agent self-adaptive tasks. In Proceedings of the Twenty-fourth AAAI Conference on Artificial Intelligence and the Twenty-second Innovative Applications of Artificial Intelligence Conference: July 11-15, 2010, Atlanta, Georgia, 1702-1709. Menlo Park, CA: American Association for Artificial Intelligence Press.

## Published Version

<http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/viewFile/1755/2285>

## Permanent link

<http://nrs.harvard.edu/urn-3:HUL.InstRepos:9962004>

## Terms of Use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Open Access Policy Articles, as set forth at <http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#OAP>

## Share Your Story

The Harvard community has made this article openly available.  
Please share how this access benefits you. [Submit a story](#).

[Accessibility](#)

# Biologically-Inspired Control for Multi-Agent Self-Adaptive Tasks

Chih-Han Yu and Radhika Nagpal

School of Engineering and Applied Sciences  
Wyss Institute for Biologically-Inspired Engineering  
Harvard University

chihanyu@gmail.com, rad@eecs.harvard.edu

## Abstract

Decentralized agent groups typically require complex mechanisms to accomplish coordinated tasks. In contrast, biological systems can achieve intelligent group behaviors with each agent performing simple sensing and actions. We summarize our recent papers on a biologically-inspired control framework for multi-agent tasks that is based on a simple and iterative control law. We theoretically analyze important aspects of this decentralized approach, such as the convergence and scalability, and further demonstrate how this approach applies to real-world applications with a diverse set of multi-agent applications. These results provide a deeper understanding of the contrast between centralized and decentralized algorithms in multi-agent tasks and autonomous robot control.

## Introduction

In this paper, we summarize our recent results on the development of a bio-inspired control framework for multi-agent tasks (Yu and Nagpal 2008; 2009). Our original inspiration was the robust and scalable intelligent behavior of biological groups, e.g., bird flocking, achieved through the distributed actions of many independent agents. In these systems, each agent acts autonomously and interacts only with its neighbors, while the global system exhibits coordinated behavior. Modern multi-agent systems, such as distributed robot systems and sensor networks, are similar to these biological systems in that their overall tasks are achieved by coordinating independent agent actions. Inspired by this connection, we propose a decentralized framework for multi-agent systems to achieve coordinated tasks in a scalable, robust and analyzable manner. We summarize both theoretical and practical contributions and discuss lessons learned from this study. Based on our results, we can provide concrete statements regarding the strengths, limitations, and scope of this class of decentralized approaches and the manner in which one can leverage their strengths to interface with other autonomous control strategies in Artificial Intelligence.

Our framework addresses multi-agent self-adaptive tasks in which agents utilize their distributed sensors and actuators to autonomously solve tasks and cope with environmental changes. We formulate this problem more generally as distributed constraint maintenance on a networked agent

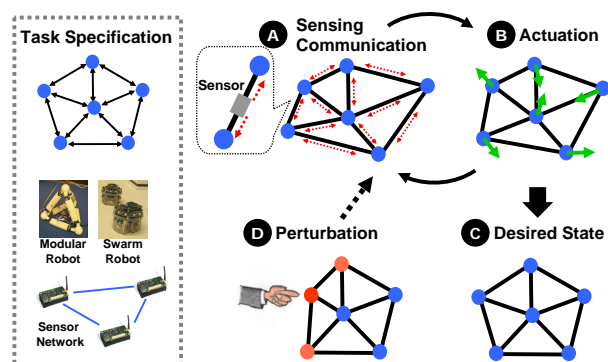


Figure 1: Concept diagram. A node indicates an agent and an edge indicates a neighborhood link. The global task is specified by inter-agent relationships, and this method can be used to describe tasks in many systems, e.g., modular and swarm robots and sensor networks. Each agent iteratively senses and communicates with neighbors (A) and performs actuation to change its state (B) until all agents' local constraints are satisfied and the desired state is reached (C). When the system is perturbed, agents restart the process (D).

system such that it can capture various multi-agent tasks. In our control algorithm, each agent aggregates information from its neighbors and then uses this information to control its state (Fig. 1). Because all agents act on local information in parallel and can potentially affect each other, two key issues are whether the whole system will converge to the desired global state and what factors determine the convergence rate.

We prove the convergence of this approach and show how the convergence rate depends on (a) the number of agents and agent topology, (b) the complexity of the task, and (c) the initial state of the agents. Using these results, we can provide precise statements on how the approach scales and how quickly agents can adapt to perturbations. These results provide a deeper understanding of the contrast between centralized and decentralized multi-agent algorithms. In addition to theoretical results, we summarize our work in applying this framework to modular robots, a type of robot that is composed of many independent and interconnected modules (agents), and other new application areas. We demonstrate that our approach allows modules to cooperatively achieve

various tasks, such as forming self-adaptive structures and manipulators. We further describe how it applies to a wider area of applications like modular robot locomotion, climbing, and adaptive orthotics. In both theoretical and practical aspects, we show that this decentralized approach outperforms centralized tree-based algorithms in dynamic environments. Our work can also serve as an example of translating a biological concept into an effective algorithmic approach.

We describe the potential impact of this work in different areas of AI. In multi-agent systems, an important question is whether one should use a decentralized or a centralized approach. Using our results, we can compare the tradeoffs and identify scenarios in which a decentralized method is advantageous. In autonomous robot control, centralized planning approaches are usually considered separately from decentralized approaches. Our study contributes to a deeper understanding of the limitations and strengths of decentralized approaches, and one can potentially combine this method with a centralized planner and leverage strengths from both approaches.

## Multi-Agent Model

In our multi-agent model, the whole system is composed of a network of agents, and the global task is described by inter-agent constraints. Each agent iteratively senses the local environment, communicates with its neighbors, and performs action until its local constraints are satisfied. The control law that each agent executes is simple, while the emerging global behaviors are sophisticated and robust. We now describe the multi-agent model and the assumed agent capabilities:

**Agent:** An agent is defined as a unit that has independent *computation, communication, sensing, and actuation* capabilities, and we denote agent  $i$  as  $a_i$ . We use a modular robot bridge, illustrated in Fig. 2 (A), as an example: Each pillar of the bridge has an independent computational unit, linear actuation to change its height, and local communication capabilities; thus, it is an agent.

**Coordination Graph:** Agents have only a one-hop local view, and they achieve global tasks by coordinating with their immediate neighbors. We represent coordination as a graph  $G$  in which the vertices represent agents and the edges represent coordination between an agent and its neighbors. A sample topology is illustrated in Fig. 1. The neighbor relationship between agents is symmetric, so the edges in  $G$  are undirected. The agent coordination graph of the modular robot bridge is a grid graph (shown in the right-side diagram of Fig. 2 (A)).

**Task Specification:** We assume that the global task can be described as inter-agent states or sensory relationships. For example, in Fig. 2 (A) there is a sensor measuring the tilt relative to the ground between each pair of neighboring agents, and one example task for the robot bridge is to maintain a level surface, which means that the task can be specified by inter-agent sensory relationship: all tilt sensor readings equal zero. This task specification scheme can be viewed generally as a constraint-based specification. The constraint between neighboring agents is satisfied when the desired inter-agent relationship is reached. Such a task specification scheme can be applied in many multi-agent tasks

that use sensor-actuator networks as their underlying architectures, e.g., modular robot locomotion, sensor network time synchronization, and multi-robot formation tasks.

## Control Algorithm

Our approach is formulated as a process by which a network of agents comes to a state of satisfied constraints by communicating only with *neighbors*. At each time step, each agent iteratively updates its new state according to feedback from its neighbors. Our approach is inspired by the biological process of consensus decision-making using only local interactions among decentralized animal groups (Couzin et al. 2005). We generalize this concept, and the agent control law can be formally written as:

$$x_i(t+1) = x_i(t) + \alpha \sum_{a_j \in N_i} f_j(\cdot) \quad (1)$$

where  $x_i(t)$  denotes agent  $a_i$ 's state at time  $t$ ,  $N_i$  denotes the set of  $a_i$ 's neighbors,  $0 < \alpha < \frac{1}{|N_i|}$  is a damping factor that determines the degree of reaction to each neighbor's feedback, and  $f_j(\cdot)$  represents the feedback function from neighbor  $a_j$ .

There are two forms of feedback functions:

- (i)  $f_j(\cdot) = x_j(t) - x_i(t) - \Delta_{ij}^*$ , which addresses scenarios in which the agent sensor and actuation states are the *same* and  $\Delta_{ij}^*$  is the desired state difference between  $a_i$  and  $a_j$
- (ii)  $f_j(\cdot) = g(\theta_i, \theta_j) - \theta_{ij}^*$ , which extends this control law to a wider range of scenarios in which the agent sensor and actuation states can be *different*.  $g(\theta_i, \theta_j)$  represents feedback agent  $a_i$  receives from  $a_j$ , and  $\theta_{ij}^*$  is the desired sensory difference between  $a_i$  and  $a_j$ .

In our terrain-adaptive bridge example, we use a feedback function of form (ii):  $f_j(\cdot) = \theta_{ij}$ , where  $\theta_{ij}$  indicates the tilt sensory reading between agent  $a_i$  and  $a_j$  and  $\theta_{ij}^* = 0$  (because we want to achieve a level surface).

We outline three sufficient conditions for designing feedback function  $f_j(\cdot)$  that guarantee correctness of the control law (all agents will complete the desired task). These conditions can guide us in efficiently designing controllers for various multi-agent systems that can be viewed as sensor-actuator networks:

$$g(\theta_i, \theta_j) - \theta_{ij}^* = 0 \Leftrightarrow \theta_j - \theta_i = \theta_{ij}^* \quad (2)$$

$$\text{sign}(x_j(t) - x_i(t) - \Delta_{ij}^*) = \text{sign}(g(\theta_i, \theta_j) - \theta_{ij}^*) \quad (3)$$

$$g(-\theta_i, -\theta_j) = -g(\theta_i, \theta_j) \quad (4)$$

Intuitively, condition 1 (Eq. 2) means that  $g$  only “thinks” that the system is solved when it actually “is” solved; condition 2 (Eq. 3) means that when not solved, each sensory feedback  $g(\theta_i, \theta_j)$  at least points the agent *in the correct direction* to satisfy the local constraint  $\theta_{ij}^*$  with a neighboring agent  $a_j$ ; and condition 3 (Eq. 4) means that  $g$  is *anti-symmetric*. To utilize the proposed agent control law to solve tasks on different systems, we must address the following challenge: we need to appropriately design the function  $g$  such that the above three conditions are satisfied. We will show the convergence proof and various example applications in the following sections. We also note that both

feedback functions (i) and (ii) for our agent control law presented here are linear functions. One interesting area for future work is to extend this linear feedback form to more generalized perception-action relationships.

### Convergence Analysis

Our approach is decentralized with many agents acting on local information in parallel. A key question that arises is whether these local actions will always produce the desired global state (convergence from all initial states). Furthermore, if the system will converge, what is the convergence speed? The key point of our analysis is the aggregation of all agent control laws as a linear dynamical system, which allows us to study the emerging global behavior of all agents by analyzing a single system. By leveraging results from spectral graph theory and stochastic matrix properties, we prove that the system will converge to the desired state for arbitrary connected coordination graphs and goals. Here we first show results for the control law with feedback function (i) in static agent topologies.

We first aggregate all agent update equations to become collective dynamics. Let  $X(t) = (x_1(t), x_2(t), \dots, x_n(t))'$ . Based on Eq. 1, we can write the collective dynamics of all agents as:

$$X(t+1) = A \cdot X(t) + \tilde{b} \quad (5)$$

where  $A$  is an  $n \times n$  matrix that models the agents' interaction dynamics and  $\tilde{b}$  is a bias vector that depends on the task considered. If  $\tilde{b} = \vec{0}$ , this equation is equivalent to a distributed consensus (DC) (Olfati-Saber, Fax, and Murray 2007). Our key result is that the inclusion of a nonzero bias vector does not affect the convergence analysis, so the convergence analysis of DC applies here as well. Let  $\mu_2(A)$  be the second largest eigenvalue of  $A$ ; we can then prove the following theorem:

**Theorem 1 (Convergence)** *Let  $X^*$  be the desired state in which the global task is completed. If agent graph  $G$  is connected, the agents will converge to  $X^*$  for all initial conditions at an exponential rate  $\mu_2(A)$  and  $0 \leq \mu_2(A) < 1$ .*

The error tolerance  $\epsilon$  represents the fact that agents have finite resolution in controlling their actuation and that some level of inaccuracy must therefore be tolerated.

In the case of feedback function (ii), the generating matrix and the biased vector become *time varying* (denoted as  $A(t)$  and  $\tilde{b}(t)$ , respectively). The collective dynamics are then:

$$X(t+1) = A(t) \cdot X(t) - \tilde{b}(t) \quad (6)$$

To prove convergence, we use the property that the infinite product of *row stochastic* and *symmetric* matrices (Wolfovitz 1963). In fact,  $A(t)$  is guaranteed to be row stochastic and symmetric for all  $t$  if all of the conditions for  $g$  outlined in Eq.2 – Eq.4 are satisfied. This allows us to prove convergence for the control law with feedback function (ii):

**Theorem 2 (Convergence)** *Let feedback function (ii),  $f_j(\cdot)$ , satisfy Eq.2 – Eq.4 and agent graph  $G$  be connected. The agents will converge to  $X^*$  for all initial conditions with an exponential rate at least:  $\mu_2^* = \max_t \mu_2(A(t))$ .*

### Factors Affecting Performance

We now address three important questions: 1) Scalability and impact of topology: How is the convergence time affected by the number of agents and coordination graph structure? 2) Effect of tasks: How do different tasks affect the convergence time? 3) Reactivity: How do the agents react to perturbations from the desired state? Here, we show analytical results that provide precise answers to these questions. We present analysis for feedback functions of type (i), though the same procedure also applies to feedback functions of type (ii).

Before proceeding to analysis, we first derive an inequality for the convergence time, which can be defined as the number of iterations required to achieve the desired task within a certain error tolerance. We define: (1)  $Y(t) = \|X(t) - X^*\|$ , and (2)  $X(t)$  is  $\epsilon$ -approximation of  $X^*$  if  $Y(t) < \epsilon$ . We can derive the number of iterations required,  $t_{\max}$ , to achieve  $\epsilon$ -approximation:

$$t_{\max} \leq \log_{\mu_2(A)} \left( \frac{\epsilon}{\|Y(0)\|} \right) \quad (7)$$

**1) Scalability and impact of topology:** We can see from Ineq. 7 that the number of iterations required for convergence,  $t_{\max}$ , depends on  $\mu_2(A)$ . We can further decompose  $A = I - \alpha L$ , where  $L$  is *Laplacian matrix*, and  $\alpha$  is the same damping factor in Eq. 1. The second eigenvalue of  $L$  is the algebraic connectivity, which encodes how well the graphs are connected. While algebraic connectivity has been studied extensively in graph theory, its use in understanding decentralized algorithms is relatively new. Using the connection between  $A$  and  $L$  and results from (Mohar 1991), we prove bounds on  $\mu_2(A)$  that describe how the algorithm scales.

**Theorem 3 (Scalability)** *Let  $D$  be the graph  $G$ 's diameter and  $n$  be the number of agents. Then  $\mu_2(A) \leq 1 - \frac{4\alpha}{n \cdot D}$ .*

This shows us how the worst-case convergence rate scales with the number of agents and agent graph diameter. One can also directly compute  $\mu_2(A)$  for a given topology, providing a more precise prediction of the convergence time.

**2) Effect of tasks:** As we can see from Ineq. 7,  $t_{\max}$  increases logarithmically with  $Y(0)$ , which implies that the convergence time changes slowly as the deviation  $Y(0)$  increases. The following theorem indicates that if the agent topology and initial states are known, we can calculate the maximal number of iterations required to achieve *any* task.

**Theorem 4** *Assuming that the agents' initial states are known and  $C = \sum_i x_i(0)$ , the number of iterations required to achieve  $\epsilon$ -approximation for any task is at most:*

$$t_{\max} = \lceil \log_{\mu_2(A)} \left( \frac{\epsilon}{\sqrt{2}C} \right) \rceil$$

**3) Reactivity:** Another important question is how the system reacts to perturbations. From Ineq. 7, we see that if  $Y(0)$  is small, then only a few iterations will be needed to achieve  $\epsilon$ -approximation. In our analysis (Yu and Nagpal 2008), we varied the system size dramatically and discovered that the system's reactivity toward small perturbations scale extremely well with the number of agents (only a few

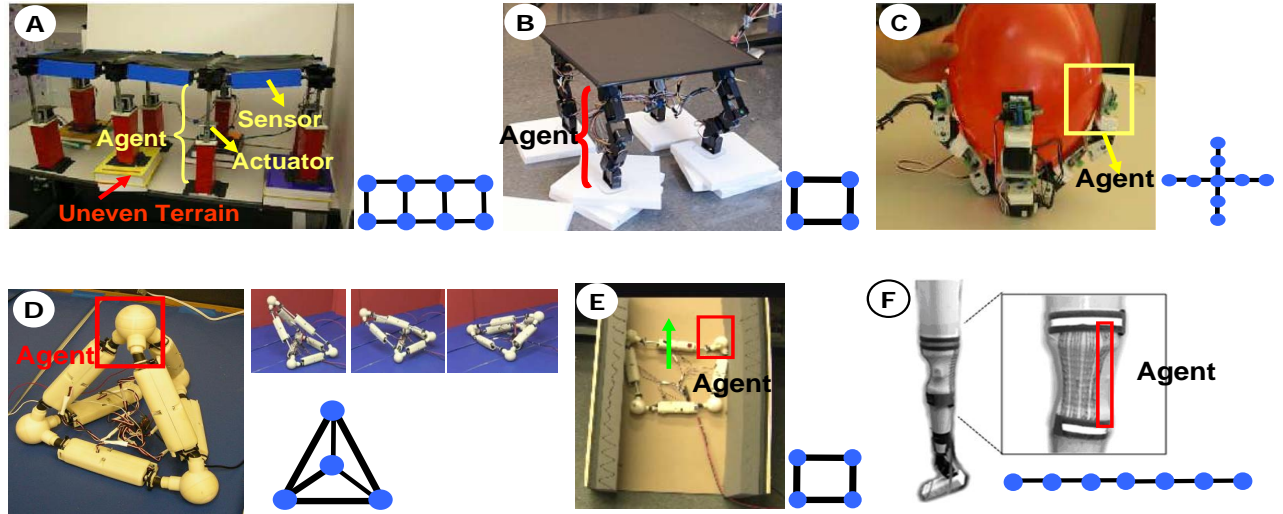


Figure 2: Many modular robot tasks can be solved with this framework, including formation of an adaptive structure (A-B), pressure-adaptive grasping (C), locomotion (D), climbing (E), and forming adaptive orthotics (F). These systems can be viewed more abstractly as sensor-actuator networks and their agent coordination graph topologies are shown on the right.

iterations are required) in both regular and irregular topologies. If the environment changes smoothly, large changes will appear as small perturbations over time. This explains why the algorithm performs particularly well at tasks that require constant adaptation.

### Relationships to a Broader Context of AI Multi-agent Applications

Our approach is reactive and well-suited for tasks that require constant adaptation under dynamic conditions. Many tasks that require agents to adapt their strategies based on changing environments share this requirement. We now discuss how the above results can be applied to various examples of AI problems, some of which we have already demonstrated.

*Environmentally-Adaptive Structures:* We assembled modules to form a terrain-adaptive bridge and a self-balancing table (Fig. 2 (A-B)). In these structures, each leg is an agent, and there is a tilt sensor mounted between each pair of neighboring agents. The coordination graphs are shown to the right of the figures. The specified task is for each agent to maintain zero tilt angles with respect to all of its neighbors. Each agent iteratively uses its tilt sensor feedback to control its state (leg length) by executing Eq. 1. Our results show that agents can keep the bridge and table surfaces level, even when the underlying terrain is constantly changing.

*Adaptive Modular Gripper:* When modules are equipped with pressure sensors and motors, they can form a gripper that can reconfigure itself to manipulate a fragile object using distributed sensing and actuation (Fig. 2 (C)). In this task, each module is specified to maintain equal pressure on its neighbors. Without predetermining the grasping posture, the modules are able to collectively form a grasping configuration that conforms to the shape of the object. When the gripper is perturbed by an external force, its grasping

posture autonomously adapts to maintain the equal pressure state, similar to how the human grasp adapts to sudden external impacts without careful contemplation. One future application along this line is an adaptive orthotic that can actuate and reconfigure its shape to apply the optimal force to correct a patient’s gait (Fig. 2 (F)).

*Adaptive Locomotion and Climbing:* In the previous examples, modules were programmed to achieve a single self-adaptive task. We extended this framework to solve locomotion tasks with a sequence of self-adaptive tasks. In modular tetrahedron rolling locomotion (Fig. 2 (D)), each locomotion step is modeled as a pressure-adaptive task, and agents autonomously start the next locomotion cycle as soon as inter-agent pressure constraints are achieved. This allows the robot to move adaptively on different slopes. This approach can potentially be applied to various other autonomous locomotion tasks, e.g., for a modular robot to achieve vertical climbing (Fig. 2 (E)).

*Experimental Results:* We also conducted various experiments to evaluate several different aspects of the multi-agent control approach, including the following: (1) its capacity to adapt to external perturbations as well as internal faults; (2) how different initial conditions and different robot configurations would affect the time required to complete the desired tasks; and (3) scalability of the number of modules. Detailed experimental results are presented in (Yu and Nagpal 2009), and here we show one set of experiments that evaluate the modular gripper’s (Fig. 2 (C)) capability to adapt to external perturbations. We define  $\epsilon = \frac{Y(t)}{Y_{\max}}$  as a ratio that measures how far agents are from the desired state, where  $Y_{\max}$  is the maximal possible perturbation. Fig. 3 shows  $\epsilon$  vs. time as the gripper encounters four different perturbations. We can see that  $\epsilon$  decreases to less than 3% after 50 – 70 iterations in each case. This result shows that our decentralized control law can efficiently lead agents to



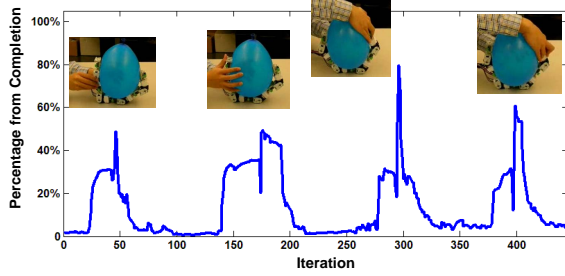


Figure 3: We constantly perturb the gripper by applying force. The robot is able to re-adapt after being perturbed.

recover from exogenous perturbations.

The robot hardware experiments and simulations we conducted in (Yu and Nagpal 2009) can be summarized:

- The control law allows us to easily program agents to achieve complex and correct global behaviors.
- This approach is inherently error-correcting and able to accomplish tasks robustly even when agents are equipped with noisy sensors and actuators.
- A diverse set of multi-agent applications can be abstractly viewed as the same task and tackled with this framework.
- The theoretical results presented in the previous section coincide with our empirical results, thereby validating our assertion that the theories developed here can be used to estimate real-world system performance.

### Centralization vs. Decentralization

An important question in networked multi-agent systems is whether one should use a decentralized approach, such as the method described here in which agents iteratively communicate and react to arrive at a solution, or use a centralized tree-based approach in which a root agent collects all of the information from other agents. This question is relevant to a wide range of multi-agent applications. Using our results, we can describe the tradeoffs between these two approaches.

For the centralized approach, we assume that a root agent collects all of the information from all agents using a spanning tree, computes a final state for every agent, and then disseminates the results back to each agent. This incurs two costs: (a) a communication cost for collecting/disseminating information, and (b) a computation cost for the root node. In most homogenous multi-agent systems, each agent has fixed communication and computation power. For the kinds of tasks considered here, communication is often a more severe bottleneck: if an agent can only collect a constant amount of information per unit time, then the time needed to collect all agents' states is  $O(n)$  ( $n$ : number of agents) and not  $O(D)$  ( $D$ : diameter). This cost is paid for every task, regardless of the distance between the initial and desired states. This results in poor reactivity in the case of small perturbations, where information must travel all of the way to the root agent before the state can be resolved. In contrast, the communication cost of the decentralized algorithm is  $O(t_{\max})$ , which depends on both the topology and the distance from the goal. The relationship between topology and performance is worse than in the centralized case in

some cases. However, if the distance from the goal is small, e.g., a small perturbation or a slowly changing environment, then the system reacts rapidly in only a few iterations, even when  $n$  is large.

This suggests that while decentralized algorithms may pay a significant start-up cost to achieve a steady state, they are extremely reactive to perturbations. Thus, they are more appropriate when the goal is to maintain constraints over long periods of time under uncertain and changing conditions, rather than to produce a solution once.

The decentralized strategy also allows these multi-agent systems to be more robust because there is *no* single agent that plays a critical role in the group task and that can consequently become a potential point of failure for the system. Instead, all agents play the same role and execute the same control law. This allows one failed agent's function to be easily replaced by another agent. Because all agents interact only with local neighbors, each failed agent tends to only affect its local area. In (Yu 2010), we prove the robustness properties of this approach in the face of communication and actuation errors. Our results show that this approach allows agents to achieve the desired task even if the communication links between agents fail temporarily, provided that the time-varying agent coordination graph satisfies the periodically-connected definition. While encountering a single agent actuation error, we prove that all other agents will try to accommodate this failed agent such that the desired task is still eventually achieved.

### Centralized Planning vs. Decentralized Control

While the type of decentralized approaches presented here can scale to a vast number of agents, the task representation scope is usually more constrained than that of centralized planners, which can potentially express a wider range of tasks across a longer time horizon. In contrast, the main limitation to centralized planning is its scalability in degrees of freedom, e.g., the number of agents or number of actuators, that it can coordinate simultaneously (also referred to as the "curse of dimensionality"). Although many approximate centralized planning strategies have been developed to address such a challenge, they are usually difficult to implement and are constrained to specific task domains.

Biological systems have evolved to intelligently leverage the strengths of both approaches. For example, human beings use their brains to plan sophisticated tasks, e.g., arranging daily schedules, in a centralized manner, while millions of cells regulate our body in a decentralized way. It would be interesting to explore a mixed strategy that is composed of both centralized planner and decentralized controllers. We see that the type of decentralized strategies considered here shows strengths in tasks that require constant adaptations to a changing environment. This observation helps us to identify scenarios that can be solved more effectively with a distributed approach. One can then design a centralized planner that oversees and coordinates decentralized tasks. For example, a humanoid robot utilizes a centralized planner to plan a trajectory to reach an object, and decentralized controllers actuate a gripper that exploits distributed sensors to grasp the object and self-adapt to sudden external impacts.

## Related Work

Several groups have designed decentralized algorithms for multi-agent systems, such as modular robots (Yim, M. et. al 2007), swarm robots (Dorigo et al. 2004), and sensor networks (Lucarelli and Wang 2004; Akyildiz et al. 2002). While various decentralized approaches have been proposed, most approaches lack theoretical treatment; furthermore, most approaches are specialized for particular tasks and difficult to generalize to other tasks or configurations. In this work, we present a comprehensive framework for analyzing the important properties of a class of decentralized algorithms. We also show that an in-depth theoretical analysis allows us to identify the scope of this approach and to further generalize it to various multi-agent applications.

On the other hand, nearest neighbor rules have been widely studied in control theory, including in consensus problems (Olfati-Saber, Fax, and Murray 2007; Bertsekas and Tsitsiklis 1989), flocking (Jadbabaie, Lin, and Morse 2002), and formation control (Fax and Murray 2004). We bridge analytical results from these areas and further expand their application domains by generalizing such control laws (feedback function (ii)) to capture a wider range of systems that can be viewed as sensor-actuator networks. Our task specification also has similarities to distributed constraint optimization (DCOP) (Yokoo 2001). The main distinction is that DCOP uses discrete variables for agent states, while agent states in this framework are continuous variables that represent control parameters.

## Perspectives, Discussions, and Conclusions

We have presented and analyzed a class of decentralized algorithms for multi-agent tasks. We proved the convergence properties and characterized how system size, topology, and initial state can affect the performance. Several valuable lessons were learned from this study. First, in decentralized agent systems, it can be tedious to develop cooperation mechanisms that allow agents to integrate dynamic environment information and act cooperatively. We have shown here that bio-inspired control laws can be simple and effective alternatives to achieve such goals. One challenge along this direction is how to effectively discover principles that govern biological systems and translate them into algorithms. More systematic ways of extracting control laws from living systems are essential to achieve this goal. Second, an important criterion for decentralized approaches is convergence to the goal state. If we can model interactions among agents in a matrix or another mathematical form, we can apply rich theories to rigorously analyze such systems.

Third, many existing robotic systems can be framed as multi-agent systems if we interface each component with the appropriate capability, typically sensing. These systems can exploit decentralized agent strategies, thereby acquiring the associated scalability and robustness. Finally, in this framework, we assume all agents play equally important roles and receive sensory information of the same importance. In spatially-distributed agent groups, some agents may obtain privileged information based on their locations and need to play a more important role in the group task. In (Yu, Werfel, and Nagpal 2010), we propose the concept of *implicit*

*leaderships* that allow better-informed agents to effectively influence group decisions based on local interactions. One interesting topic in this direction is the design of an approach to incorporate more complex local agent information while preserving simplicity of the decentralized agent control.

## Acknowledgement

This research is supported by an NSF EMT Grant (CCF-0829745) and Wyss Institute for Bio-inspired Engineering.

## References

- Akyildiz, I. F.; Su, W.; Sankarasubramaniam, Y.; and Cayirci, E. 2002. Wireless sensor networks: a survey. *Computer Networks* 38:393–422.
- Bertsekas, D. P., and Tsitsiklis, J. 1989. *Parallel and Distributed Computation*. Prentice Hall.
- Couzin, I.; Krause, J.; Franks, N.; and Levin, S. 2005. Effective leadership and decision making in animal groups on the move. *Nature* 433.
- Dorigo et al. 2004. Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots* 17(2-3):223–245.
- Fax, J. A., and Murray, R. M. 2004. Information flow and cooperative control of vehicle formations. *IEEE Trans. on Auto. Control* 49(9):1465–1476.
- Jadbabaie, A.; Lin, J.; and Morse, A. S. 2002. Coordination of groups of autonomous agents using nearest neighbor rules. *IEEE Trans. on Auto. Control* 48(6):988–1001.
- Lucarelli, D., and Wang, I. 2004. Decentralized synchronization protocols with nearest neighbor communication. In *Proc. Sensys*.
- Mohar, B. 1991. Eigenvalues, diameter, and mean distance in graphs. *Graphs and Combinatorics* 7:53–64.
- Olfati-Saber, R.; Fax, J.; and Murray, R. 2007. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* 95(1):215–233.
- Wolfowitz, J. 1963. Products of indecomposable, aperiodic, stochastic matrices. *Proc. Am. Math. Soc.* 15:733–736.
- Yim, M. et. al. 2007. Modular self-reconfigurable robot systems – challenges and opportunities for the future. *IEEE Robotics and Automation Magazine* 43–53.
- Yokoo, M. 2001. *Distributed constraint satisfaction: foundations of cooperation in multi-agent systems*. Springer-Verlag.
- Yu, C.-H., and Nagpal, R. 2008. Sensing-based shape formation tasks on modular multi-robot systems: A theoretical study. In *Proc. AAMAS*.
- Yu, C.-H., and Nagpal, R. 2009. Self-adapting modular robotics: A generalized distributed consensus framework. In *Proc. ICRA*.
- Yu, C.-H.; Werfel, J.; and Nagpal, R. 2010. Collective decision-making in multi-agent systems by implicit leadership. In *Proc. AAMAS*.
- Yu, C.-H. 2010. *Biologically-Inspired Control for Self-Adaptive Multiagent Systems*. Ph.D. Dissertation, Harvard University.